

Responsável: Onivaldo Rosa Junior

PPU - Pacote Padrão UNA-SUS

Especificação de empacotamento

Versão preliminar

Sumário

Sumário	1
1- Recursos educacionais multiuso.....	2
2- Conteúdo HTML5	4
3- Descritor	5
Instruções para descrição:	6
Variáveis PACOTE C10	8
Variáveis PACOTE C11	8
Variável obrigatória STATUS:	8
Variável recomendada VIDEO:	9
Variável recomendada NAVIGATION:	9
Variável recomendada DOWNLOAD:	9
Variável recomendada EXTERNAL_LINK:.....	10
Variável recomendada ANSWER:	10
4- API.....	11
5- Player de teste.....	14

1- Recursos educacionais multiuso

O **Pacote Padrão UNA-SUS - PPU**, surgiu como uma necessidade de padronizar recursos educacionais complexos e multimídia, otimizados para utilização e distribuição, e com grande capacidade de visualização e reutilização, e que necessitavam de persistência para geração de lógicas de navegação e avaliação formativa.

Indagações que nortearam o desenvolvimento do padrão:

- Recurso educacional autocontido capaz de ser armazenado em acervos.
- Metadados que permitam a geração automatizada de pacotes derivados para uso específico, exemplo, dispositivos móveis, mesmo em modo *off-line*.
- Mecanismos simples e objetivos de rastreamento e análise de uso do recurso.

O **PPU** foi desenvolvido e testado pela SE/UNA-SUS buscando responder estas indagações e utiliza fortemente o padrão **HTML5** e tecnologias correlatas.

Inicialmente, o pacote poderia rodar direto em um navegador sem servidor, mas com as novas **diretrizes de segurança** dos navegadores isso ficou problemático. Atualmente somente o Navegador Mozilla Firefox é capaz de rodar o pacote nestas condições. Utilizando para isto um protótipo de *player* WEB capaz de rodar direto do disco, com a tecnologia **localstorage** para persistência de dados.

O *player* indicado acima deve ser utilizado apenas como demonstração tecnológica e para fins de testes locais de pacotes durante o desenvolvimento dos mesmos.

Depois de um período de testes optou-se por um modelo de *APIs* que permitissem um pacote multiuso, acionado por **players on-line e off-line**, e com características que facilitassem a **integração** com AVAs e LMSs.

Adotou-se para a definição de score geral no pacote valores em ponto flutuante entre 0 e 1, visando a integração direta com outras aplicações (por exemplo, moodle) via padrão LTI.

O **Acervo de Recursos Educacionais em Saúde - ARES** permitirá o armazenamento destes pacotes, e seu acesso interativo por meio de um *player padrão*, assim como a distribuição dos pacotes a outros sistemas.

A especificação do PPU, define o empacotamento dos recursos em um arquivo ZIP, cujo nome deve ser composto pelos atributos de persistência indicados no descritor do pacote, separados por “_”, exemplo: SE_UNASUS_0002_HANSENIASE.zip

O Pacote é composto por três elementos principais:

- Conteúdo desenvolvido em HTML5 e tecnologias correlatas, com ponto de entrada definido pelo arquivo **index.html**, que deve estar na raiz do pacote, todos os elementos fundamentais para a execução do pacote devem estar no mesmo. Instruções mais detalhadas são indicadas no item 0.

- Descritor do pacote, em formato JSON padronizado pela SE/UNA-SUS, arquivo **se_unasus_pack.json** que deve estar na raiz do pacote. Instruções de como descrever o pacote são indicadas no item 0.

- API de conteúdo, desenvolvida e disponibilizada pela SE/UNA-SUS, arquivo **se_unasus_pack.js** que deve estar na raiz do pacote. Instruções de como utilizar esta API são indicadas no item 0.

O padrão também exige que o pacote tenha uma imagem de **“capa”** para o pacote, no formato PNG, com 1280x720. Esta deve estar a raiz do pacote, com o de arquivo conforme indicado no descritor do pacote

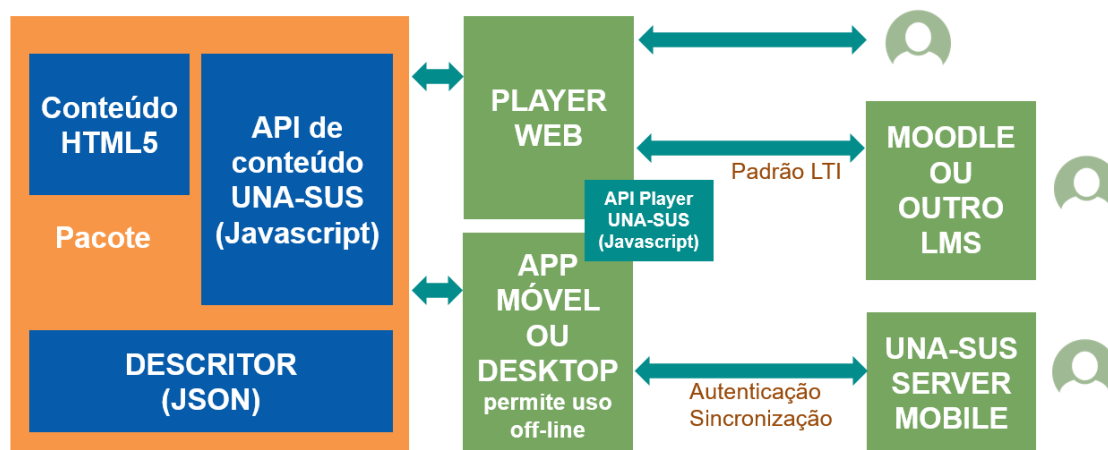
A

Figura 1 apresenta o pacote e seus elementos e a visão de três cenários de execução:

- Usuário acessando diretamente um player WEB
- Usuário acessando um ambiente LMS integrado ao player WEB via LTI
- Usuário acessando um APP móvel integrado ao servidor de aplicativos móveis da UNA-SUS

Todos os cenários foram planejados para utilizar a autenticação do ACESSO UNA-SUS, nosso **single-sign-on**

Figura 1 - Visão geral do PPU e sua contextualização em ambientes de execução.



Autenticação do Usuário em qualquer caso é feita pelo Acesso UNA-SUS (Single Sign-On mantido pela SE/UNA-SUS)

2- Conteúdo HTML5

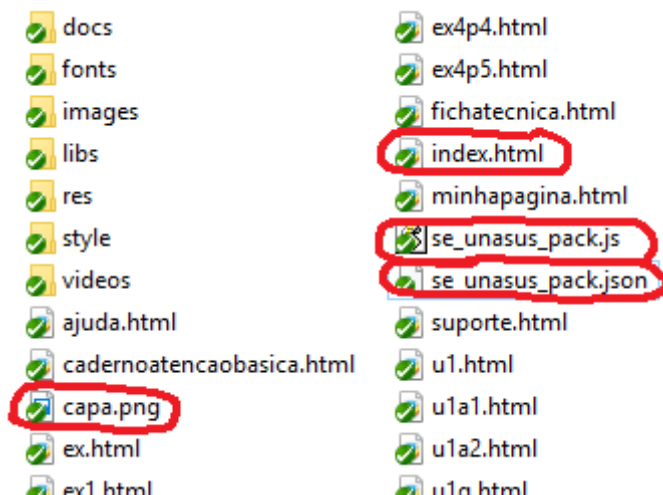
O conteúdo HTML5, deve seguir algumas regras para permitir seu uso off-line, e facilitar o processamento automático de pacotes:

- Utilizar apenas padrões abertos relacionados a HTML, CSS e Javascript.
- Não utilizar referências externas para fontes tipográficas, imagens, áudios, vídeos, arquivos PDF, legendas, transcrições entre outros.
- Links externos somente para informação não essencial.
- Utilizar bibliotecas javascript abertas, e incorporadas ao pacote.
- Imagens apenas em JPG, PNG, GIF e vetoriais em SVG.
- Vídeos em MP4, com nomes terminados com “_” e com a resolução (exemplo video1_360p, videob_720p). As resoluções utilizadas, devem obrigatoriamente seguir as instruções de nomenclatura acima, e devem ser descritas no arquivo de descrição, conforme indicado no item 0
- Áudios em MP3
- Responsividade é obrigatória para pacotes que suportem mais de um tipo de dispositivo.
- Canvas e WebGL podem ser incorporados conforme o mercado.
- Conteúdos complementares indicados por link externo, devem ser tratados para conter indicação de “**recurso on-line**”, ou seja, não funcionam em modo *off-line*.

A

Figura 2 apresenta um exemplo de organização de arquivos no pacote, apresentando os arquivos obrigatórios destacados.

Figura 2 - Distribuição exemplo de arquivos e pastas em um PPU.



A index.html e todas as páginas referenciadas de forma direta, ou seja, que serão chamadas pelo player, devem incluir o javascript da API, e inicializa-la conforme a necessidade. Maiores informações no item 0.

Na construção do conteúdo, deve-se considerar que o mesmo vai ser exibido dentro de um player, e que cuidados devem ser tomados para não sobrescrever ou gerar confusão na navegação do usuário. As referências e auxílios de navegação devem ser na medida do possível neutros, sem conter informações absolutas sobre o ambiente, e sim somente ao conteúdo. O Pacote deve ter a característica de ser autocontido.

3- Descritor

O arquivo descritor deve obrigatoriamente possuir o nome **se_unasus_pack.json** e ficar localizado na pasta raiz do pacote. Seu formato é JSON.

O arquivo descreve as seguintes informações:

"Package" => Indica a versão do pacote

"Institution" => Indica dados da instituição produtora do recurso

"Resource" => Informações sobre o recurso

"SupportedDevices" => Suporte a dispositivos

"Video" => Resolução e Formatos de Vídeo utilizados no recurso

"Persistence" => Informações para evitar conflitos entre variáveis persistentes

"Dictionary" => Dicionário de variáveis persistentes utilizadas no pacote

Um exemplo parcial deste arquivo:

```
{
  "Package": "CI1",
  "Institution": {
    "AroucaID": 4510421,
    "Acronym": "SE/UNA-SUS",
    "Name": "Secretaria Executiva da Universidade Aberta do SUS"
  },
  "Resource": {
    "Name": "Hanseníase",
    "Language": "pt-BR",
    "CreationDate": "2016-01-22",
    "Review": 1,
    "ReviewDate": "2016-02-07",
    "CoverImage": "capa.png"
  },
  "SupportedDevices": [
    "Tablet",
    "Desktop"
  ],
  "Video": {
    "Formats": [
      "mp4"
    ],
    "Resolutions": [
      "360p",
      "720p"
    ]
  },
  "Persistence": {
    "InstitutionAcronym": "SE_UNASUS",
    "Version": "0002",
    "Name": "HANSENIASE"
  },
  "Dictionary": [
    {
      "Variable": "STATUS",
      "Description": "Variável padrão, indic..",
      "Format": {
        "status": "Valor nominal te... ",
        "percentage": "Valor inteiro entre 0 e 100.. ",
        "LTIValue": "Valor ponto flutuante entre 0.. ",
        "U1A1": "Valor booleano indic.. ",
        "U1A2": "Valor booleano indic..."
      }
    }
  ]
}
```

Instruções para descrição:

Package: Deve conter o tipo de pacote, os valores possíveis são:

CI0 => identifica um pacote simples que não fará uso de persistência

Este tipo de pacote só necessita incluir a API no index.html e chamar a função de inicialização, apenas para permitir o registro mínimo de uso do pacote.

C11 => identifica um pacote padrão que fará uso de persistência, com STATUS (situação, percentual de andamento e nota)

Este tipo de pacote necessita incluir a API e inicializa-la no index.html e em todas as páginas onde a persistência será utilizada, e fazer uso destas funções para manter no mínimo a variável padrão STATUS e suas propriedades obrigatórias. (Vide item 0)

Institution: Deve conter a indicação da instituição produtora principal do pacote

"AroucaID" => Código numérico da instituição na Plataforma Arouca

"Acronym" => Sigla da instituição

"Name" => Nome completo da instituição

Resource: Deve conter informações gerais do pacote

"Name" => Nome completo do recurso

"Language" => Idioma do recurso no formato ISSO 639

"CreationDate" => Data de criação ou empacotamento do recurso no formato "AAAA-MM-DD"

"Review" => Número da revisão do recurso, caso não houver utilizar 0

"ReviewDate" => Data da última revisão do recurso no formato "AAAA-MM-DD", se não houver revisão repetir a data de criação.

"CoverImage" => Nome do arquivo de capa do recurso, exemplo "capa.png"

SupportedDevices: Deve conter informações sobre o suporte a classe de dispositivos é indicada na forma de um array, com a indicação das classes suportadas. As classes disponíveis para catalogação são: "Smartphone", "Tablet" e "Desktop"

Video: Deve conter informações sobre os vídeos disponibilizados, formato e resolução.

"Formats" é um array com os tipos, indicado pela extensão do arquivo, hoje recomendamos apenas utilizar mp4.

"Resolutions" é um array com resoluções, recomendamos "360p" e "720p"

Os vídeos devem estar com a nomenclatura indicada de forma padronizada, exemplo "apresentação_jose_720p.mp4"

Persistence: Deve conter informações básicas para gerar o nome do pacote e o prefixo interno (basename) para evitar conflito de variáveis entre pacotes. Devem ser utilizados apenas caracteres entre A e Z e 0 e 9, com permissão de uso do separador "_", utilize apenas caixa ALTA, e não utilize caracteres fora da faixa indicada.

"InstitutionAcronym" => Sigla da instituição respeitando as regras de caracteres

"Version" => Versão do recurso, entre 0001 e 9999, 0000 pode ser utilizado apenas para fins de desenvolvimento e testes.

"Name" => Nome principal do pacote

Dictionary: Deve conter informações que permitam a um analista externo conhecer a natureza de cada variável de persistência utilizada pelo pacote.

"Variable" => Nome da variável, utilize apenas A-Z 0-9 "_"
"Description" => Descrição da variável
"Format" => JSON indicando o formato da variável, no lugar do valor, colocar a descrição.

Exemplo:

```
"Variable": "ANSWER",
>Description": "Indica informações acerca das respostas feitas pelo usuário durante o curso",
>Format": {
>  "stage": "Valor textual que indica o questionário ou atividade que o usuário está interagindo. Pode receber os valores: U1Q, U2Q, U3Q para os quizzes das Unidades 1,2 e 3 e EX1, EX2, EX3, EX4 para os casos clínicos",
>  "questions": [
>    {"question": "Atributo inteiro que indica a questão"},
>    {"alternative": "Atributo inteiro que indica a alternativa selecionada pelo usuário"},
>    {"correct": "Atributo booleano que indica se a alternativa selecionada pelo usuário está correta"}
>  ]
>}
```

O dicionário acima representa, valores de persistência como este exemplo:

```
"stage": "EX3",
>questions": [
>  {"question": 1, "alternative": 3, "correct": true}
>  {"question": 2, "alternative": 3, "correct": false}
]
```

Variáveis PACOTE CIO

Nenhuma variável é obrigatória, este tipo de pacote é recomendado para conteúdo multimídia HTML onde não é necessário armazenar nenhum status de uso. O dicionário fica vazio no arquivo descritor.

```
"Dictionary": []
```

Variáveis PACOTE CI1

Variável obrigatória STATUS:

O uso da variável STATUS é obrigatória, com no mínimo o formato abaixo:

```
"Variable": "STATUS",
>Description": "Variável padrão, indica o status do recurso.",
>Format": {
>  "status": "Valor nominal textual do status podendo ter os seguintes valores 'attended', 'attempted', 'completed', 'passed', 'failed'",
>  "percentage": "Valor inteiro entre 0 e 100. Especificar aqui como o valor é calculado",
>  "LTIvalue": "Valor ponto flutuante entre 0 e 1 sendo incrementado de acordo com a pontuação do usuário. Especificar aqui como o valor é calculado"
>}
```

Além dos três valores obrigatórios (status, percentage, LTlvalue) na variável STATUS, outros podem ser acrescentados conforme a necessidade do pacote, normalmente valores que auxiliem nos cálculos de modo a manter a integridade dos valores.

Observação: os valores status, percentage, LTlvalue são progressivos, ou seja, dessem sempre manter o status mais elevado alcançado pelo usuário. Se o usuário navegou obtendo 70 em percentage, navegar novamente pelos mesmos lugares não reduz este valor. Assim como ter obtido LTlvalue 0.8 realizando atividades, não deve ser reduzido pelo fato do usuário refazer com desempenho inferior as mesmas atividades. Outras variáveis, como ANSWER podem guardar cada tentativa de uma atividade formativa.

Variável recomendada VIDEO:

O uso da variável VIDEO é recomendada, com no mínimo o formato abaixo:

```
"Variable": "VIDEO",
>Description": "Indica informações acerca dos vídeos reproduzidos no curso",
>Format": {
>  "resolution": "Valor textual, deve estar entre os valores disponíveis no
>  recurso, com base no arquivo de configuração se_unasus_pack.json, exemplo
>  '360p'"
>}
```

Além do valor obrigatório (`resolution`) na variável VIDEO, outros podem ser acrescentados conforme a necessidade do pacote, normalmente valores que auxiliem na manipulação ou estatística de uso de vídeos.

Variável recomendada NAVIGATION:

O uso da variável NAVIGATION é recomendada, com no mínimo o formato abaixo:

```
"Variable": "NAVIGATION",
>Description": "Indica informações acerca da navegação do usuário no recurso",
>Format": {
>  "current": "Valor textual que indica o marcador atual de navegação do
>  usuário",
>  "previous": "Valor textual que indica o marcador anterior de navegação do
>  usuário, vazio se não existir informação anterior"
>}
```

Além dos valores obrigatórios (`current` e `previous`) na variável NAVIGATION, outros podem ser acrescentados conforme a necessidade do pacote.

Variável recomendada DOWNLOAD:

O uso da variável DOWNLOAD é recomendada, com no mínimo o formato abaixo:

```
"Variable": "DOWNLOAD",
>Description": "Indica informações de acesso à arquivo para download",
>Format": {
>  "current": "Valor textual que indica o marcador de navegação do usuário
>  no momento do download",
>  "filetype": "Indica o tipo de arquivo, os valores possíveis são 'text',
>  'video', 'audio', 'image', 'other'",
>  "filename": "Valor textual que indica o nome do arquivo + extensão"
>}
```

Além dos valores obrigatórios (`current`, `filetype`, `video` e `filename`) na variável DOWNLOAD, outros podem ser acrescentados conforme a necessidade do pacote.

Variável recomendada EXTERNAL_LINK:

O uso da variável EXTERNAL_LINK é recomendada, com no mínimo o formato abaixo:

```
"Variable": "EXTERNAL_LINK",
>Description": "Indica informações acerca do acesso a recursos externos",
>Format": {
>  "current": "Valor textual que indica o marcador de navegação do usuário
>no momento do download",
>  "url": "Valor textual do link externo, indicado pelo seu endereço
>completo"
>}
```

Além dos valores obrigatórios (`current` e `url`) na variável EXTERNAL_LINK, outros podem ser acrescentados conforme a necessidade do pacote.

Variável recomendada ANSWER:

O uso da variável ANSWER é recomendada, com no mínimo o formato abaixo:

```
"Variable": "ANSWER",
>Description": "Indica informações acerca das respostas feitas pelo usuário
>durante o curso",
>Format": {
>  "stage": "Valor textual que indica o questionário ou atividade que o
>usuário está interagindo. Pode receber os valores: U1Q, U2Q, U3Q para os
>quizzes das Unidades 1,2 e 3 e EX1, EX2, EX3, EX4 para os casos clínicos",
>  "questions": [
>    {"question": "Atributo inteiro que indica a questão"},
>    {"alternative": "Atributo inteiro que indica a alternativa
>selecionada pelo usuário"},
>    {"correct": "Atributo booleano que indica se a alternativa
>selecionada pelo usuário está correta"}
>  ]
>}
```

Além dos valores obrigatórios (`stage`, `questions`) na variável ANSWER, outros podem ser acrescentados conforme a necessidade do pacote.

4- API

A API de conteúdo é responsável pela integração entre o conteúdo do pacote e o player onde rodará este pacote.

A implementação é realizada por meio da classe SE_UNASUS_PACK, e aparecerá para uso por meio do prefixo. unasus.pack.

Para utilizar a API é necessário inclui-la no **head** das páginas que a forem utilizar:

```
<script src="se_unasus_pack.js"></script>
```

Considerando que a página está na raiz.

Para inicializar a API, utilize a chamada: unasus.pack.initialize(), veja o retorno da função na documentação da função.

Funções disponíveis (para detalhes adicionais, veja documentação no próprio código):

Retorna status de debug da API, caso API não inicializada gera mensagem de API não inicializada.

```
unasus.pack.getDebug()
```

Retorna o status de debug da API (*true* ou *false*) ou *null* se API não inicializada.

Seta status de debug da API, caso API não inicializada gera mensagem de API não inicializada.

```
unasus.pack.setDebug(debug)
```

Parâmetro debug com valores *true* para ativar as mensagens de debug ou *false* desabilita-las.

Retorna *true* se status de debug da API foi setado ou *null* se API não inicializada.

Se API inicializada e debug ativado, envia mensagem pela API, se API não inicializada emite debug utilizando console.log

```
unasus.pack.debugMessage(str)
```

Parâmetro str com mensagem textual de debug

Retorna sempre *true*

Inicializa API, em caso de erro emite mensagens de debug e substitui o conteúdo por uma mensagem de erro.

`unasus.pack.initialize()`

Retorna *true* se inicializada ou *false* se não inicializada.

Retorna status de inicialização da API

`unasus.pack.isInitialized()`

Retorna *true* se inicializada ou *false* se não inicializada.

Retorna a configuração do pacote, carregada na inicialização da API

`unasus.pack.getConfig()`

Retorna um objeto representando o arquivo de configuração do pacote (se_unasus_pack.json) ou *null* se API não inicializada.

Retorna o basename do pacote setado na configuração, é utilizado como referência para evitar colisão de variáveis.

`getBasename()`

Retorna o *basename* do pacote ou *null* se API não inicializada.

basename = InstitutionAcronym + '_' + Version + '_' + Name + '_'

Exemplo: SE_UNASUS_0002_HANSENIASE_

Retorna a versão do player onde está rodando o pacote.

`getPlayerVersion()`

Retorna a versão do player ou *null* se API não inicializada.

Retorna o nome do player onde está rodando o pacote.

`getPlayerName()`

Retorna o nome do player ou *null* se API não inicializada.

Retorna dados do usuário, os dados mínimos são `RealName` e `Id`

`getPlayerUser()`

Retorna um objeto com as propriedades `RealName` e `Id`, representando o usuário ou `null` se API não inicializada.

Retorna o Nome completo do usuário no player

`getPlayerUserRealName()`

Retorna o *Nome completo* do usuário ou `null` se API não inicializada.

Retorna o Id do usuário no player

`getPlayerUserId()`

Retorna o *Id* do usuário ou `null` se API não inicializada.

Retorna um array com as resoluções de vídeo suportadas pelo pacote

`getVideoResolutions()`

Retorna um *array* com as resoluções de vídeo suportadas pelo pacote ou `null` se API não inicializada.

Exemplo: ["360p", "720p"]

Seta o valor de uma variável persistente (que será armazenada pelo player)

`setPersistence(item,obj)`

Parâmetro item - Nome da variável persistente

Parâmetro obj - Objeto representado o valor da variável persistente, deve permitir a conversão utilizando `JSON.stringify`

Retorna `null` se API não inicializada, `true` se a variável foi setada ou `false` se a persistência falhou.

Retorna um objeto representando o último valor de uma variável persistente (armazenada pelo player)

`getPersistence(item)`

Parâmetro item - Nome da variável persistente

Retorna um objeto representando o status atual da interação do usuário com o pacote, *null* se houve falha ou API não inicializada, *undefined* se não existe valor armazenado no player

Seta a variável persistente padronizada STATUS

`setStatus(status)`

Parâmetro status - Objeto representando o status atual da interação do usuário com o pacote, representado conforme indicado no item 3-

Retorna *null* se API não inicializada, *true* se a variável foi setada ou *false* se a persistência falhou. Esta função verifica a validade dos valores em status, percentage e LTlvalue

Retorna a variável persistente padronizada STATUS

`getStatus()`

Retorna um objeto representando o status atual da interação do usuário com o pacote, *null* se houve falha ou API não inicializada, *undefined* se não existe valor armazenado no player

Observação: *Antes de inicializar uma variável persistente, leia o valor atual, caso seja undefined poderá atribuir um valor inicial, mas em caso de erro, não atribua valor a variável, pois isso pode causar inconsistência, caso haja um valor armazenado no player.*

5- Player de teste

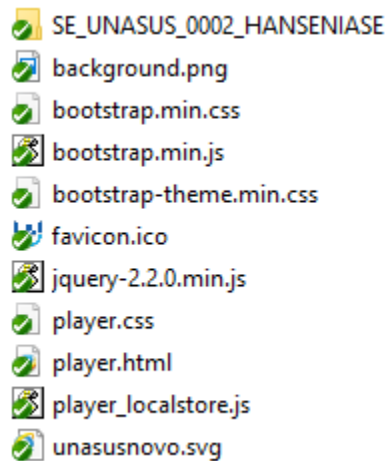
Para um teste rápido do pacote antes de compacta-lo e enviar a um player de produção, é disponibilizado pela SE/UNA-SUS, um player básico em javascript, usando localStorage para armazenamento das variáveis de persistência.

Para utilizar o player, basta colocar a pasta do pacote, ou pacotes, na raiz do player, como ilustrado na Figura 3, e alterar o arquivo `player.html`:

```
<select id="opcao_recurso" class="form-control"
onchange="selecao_recurso(this);">>
    <option value="SE_UNASUS_0002_HANSENIASE" selected>HANSENIASE</option>
</select>
```

No caso de múltiplos pacotes, criar item *option* para cada pacote.

Figura 3 - Organização de arquivos no player de teste local



O player de teste oferecerá mecanismo de seleção de qual pacote será carregado, assim como um a opção de deletar todas as variáveis persistentes do pacote, permitindo múltiplos testes. O player ativará o debug por console.

O player de teste pode ser executado diretamente do disco utilizando navegador Mozilla Firefox, ou pode ser instalado em um servidor de WEB. O arquivo de entrada é o `player.html`.

O player de teste, define como usuário: Id = 2016 e RealName = Fulano de Tal